2006 NASA SOFTWARE OF THE YEAR SUMMARY EVALUATION DOCUMENT

Identification Information	
Software Title:	**Data Parallel Line Relaxation Code (DPLR)**
NASA Case No.	**ARC-15022**
Responsible Center(s):	**Ames Research Center**

Software's Developmental Status	
Current Technology Readiness Level (1-9): 8	**Classification (A-H): A**

Significance to NASA Mission Part A - Impact on NASA's Mission

The Data Parallel Line Relaxation (DPLR) code is a computational aerothermodynamics code for the simulation, analysis, development, verification and validation of Earth and planetary entry vehicles. DPLR analysis is currently in the critical path of two of the agency's primary objectives: Shuttle Orbiter Return to Flight (RTF) and Crew Exploration Vehicle (CEV) Development. In addition, DPLR is being extensively relied on in all of NASA Earth and planetary entry missions that are currently in the design, development, operational, or post-flight assessment phase. It is used extensively in support of multiple agency priorities within all four NASA mission directorates, at two field centers.

In the Space Operations Mission Directorate (SOMD), DPLR is an enabling tool for Return to Flight activities. DPLR was used at NASA Ames Research Center (ARC) and Johnson Space Center (JSC) to define reentry aerothermal heating environments for the Shuttle Orbiter in support of the STS-107 accident analysis, the RTF Program, and STS-114 in-flight damage assessment. DPLR, in conjunction with rapid grid generation tools, enabled same day turnaround analysis of the potential entry risk of observed on-orbit tile damage, including the protruding gap filler and torn blanket, which allowed engineers to make informed decisions on whether a given damage site should be repaired prior to entry.

In support of the Exploration Systems Mission Directorate (ESMD), DPLR is a critical path tool for aerothermal and aerodynamic analysis of the CEV for the Exploration Systems Architecture Study (ESAS), the CEV Aerosciences Project (CAP), and the CEV Thermal Protection System Advanced Development Project (TPS-ADP). The entry and ascent abort convective heating portion of the CEV aerothermodynamic database is currently anchored primarily with CFD solutions generated with DPLR, and the code is also employed to plan, design, and analyze ground testing in hypersonic tunnels and arc jets. DPLR has also been coupled to a Monte-Carlo statistical analysis package and used to quantify uncertainties and sensitivities in the aeroheating environment in order to define TPS margins and reliability.

In support of the Science Mission Directorate (SMD), DPLR is currently being used to define aeroheating environments (which enables the selection and sizing of appropriate TPS materials), and assess entry risks for the Mars Phoenix and Mars Science Laboratory missions. DPLR was also the primary aeroheating tool employed during entry risk analyses of the Stardust sample return capsule (led by the Jet Propulsion Laboratory) and the Cassini-Huygens Titan entry probe (led by the NASA Engineering Safety Center and the European Space Agency). The code is used extensively within SMD for early phase mission, proposal and concept studies involving planetary entry, and technology demonstration missions.

Finally, in the Aeronautics Research Mission Directorate (ARMD), the fundamental aerodynamics hypersonics program is planning to use DPLR to anchor the development of its Physics-Based Multi-Disciplinary Analysis and Optimization design tools. DPLR is in fact an excellent example of the kind of design support tool that NASA intends to provide to industry via the fundamental aeronautics program.

Significance to Science, Technology, & Industry in General Part B – Impact on Science & Technology
While Computational Fluid Dynamics (CFD) as a field is often considered mature, hypersonic reentry aerothermodynamics is an extremely complex problem for which few commercial-off-the-shelf (COTS) tools are suitable. While there are some COTS codes that can be employed for certain low-speed entries (i.e. the Shuttle Orbiter), none demonstrate the parallel efficiency and robustness of DPLR, as demonstrated clearly during Return to Flight activities. More importantly, for higher velocity and non-Earth entries (i.e. CEV lunar return and MSL), existing COTS tools simply do not have the requisite physical models necessary to ensure accurate simulations and enable coupling to shock layer radiation and material response modeling tools as required for thermal protection system design. In addition, DPLR supports both a production environment in which a fully validated production code is being used for actual flight hardware design and simulation within three NASA mission directorates and DoD/DARPA projects, while multiple "research" versions are being used to test and benchmark new numerical and physical models for possible incorporation into the master source tree. This allows DPLR to rapidly grow and adapt as needed to meet a broader range of relevant problems. The underlying algorithms have set a standard for parallel algorithm design that continues to spur innovation at the university level. For example, the University of Minnesota has incorporated the basic algorithms in their unstructured US3D code, which has been employed recently for scramjet design and MSL parachute inflation dynamics. The robustness, generality, and execution speed of DPLR make it an ideal "training ground" for potential new state-of-the-art innovations, which may further revolutionize the field. Recent additions include automated grid alignment and integrated coupled analysis of optically thin radiation. Future plans include the addition of aeroelasticity modeling and tight coupling with radiation and material response analysis software. DPLR is having a major impact on NASA and defense aerospace industries, and can potentially benefit civilian aerospace as well. The DPLR code directly impacts the aerospace industry disciplines of aerodynamics, aerothermodynamics, and TPS design for NASA, DoD, and civilian applications. Potential applications include all civilian and military entry vehicles, hypersonic and supersonic cruise vehicles, and commercial and military launch systems. The performance and physical modeling innovations in the DPLR software have the potential to greatly enhance the design and optimization of such systems. Also, the generalized chemical kinetics and transport property packages in DPLR also make it potentially valuable for the simulation of combustion flows for both aerospace and non-aerospace applications (such as reactors or combustion engines). Other potential uses of DPLR include meteor entry analysis, breakup of de-orbiting debris (as demonstrated recently in a collaborative effort between NASA ARC and Kennedy Space Center), and missile plume signature analysis.

Significance in Impact on the Quality of Human Life Part C
NASA's stated mission is to improve life here, to extend life to there and to find life beyond. The exploration of space is of fundamental interest to the majority of Americans. Planetary entry missions are among the most popular of NASA's projects, and typically result in billions of hits on the relevant web pages. In addition, the science return from such missions directly addresses the fundamental NASA objectives of understanding the formation of the solar system and the evolution of life on Earth. The crewed exploration of space is also a centerpiece of the American space program, and continued human presence in space is of critical importance from the

standpoint of national pride and security. Each of these objectives requires that the vehicle must enter or reenter a planetary body protected only by a single point failure TPS. DPLR is a tool that improves the fidelity of such calculations over the previous state of the art, and also decreasing the turnaround time of those simulations. With the use of DPLR, the TPS designer can quantify system reliability and decrease entry risk. By better understanding the entry environment, the TPS design can be more efficient, and the mass saved can be applied to additional science instruments, increasing the intellectual impact of the mission.

DPLR has also been use to analyze de-orbiting launch debris, which can have significant environmental impact, particularly for nuclear payloads. For sample return missions, planetary protection requirements necessitate a robust TPS system enabled by DPLR analysis. From a mission safety standpoint, DPLR has already proven its utility for RTF and STS-114 analysis, and will continue to be a critical path tool for crewed entry mission safety. Analysis of missile performance and plume signature analysis can be important from a national defense standpoint.

Finally, DPLR is already installed at three universities, and the novel implicit algorithms have been used in support of several Masters and PhD level research products. The availability of a mature and easily modifiable software tool allows potential advanced degree candidates to concentrate on the physics aspects of their research, thus creating a pool of talented graduates for potential NASA, DoD, or industry careers in the field.

Extent of Current and Potential Use

DPLR is in daily use at multiple NASA and non-NASA facilities. Current use is Average, with 12 organizations and about 50 persons total using the code. Current users include:

Present Use: Government Organizations:

[1] NASA Ames Research Center (POC: Nagi Mansour, MS 230-2, Moffett Field, CA 94035. Ph. 650-604-6420)
[2] NASA Johnson Space Center (POC: Randy Lillard, 2101 NASA Rd. One, Houston, TX 77058. Ph. 281-483-6612)
[3] US Army AMRDEC (POC: James Keenan, Redstone Arsenal, AL 35898. Ph.281-876-0116)
[4] University of Minnesota (POC: Graham Candler, 107 Akerman Hall, Minneapolis, MN 55455. Ph. 612-625-2364)
[5] Pennsylvania State University (POC: Deborah Levin, 233 Hammond Bldg., University Park, PA 16802. Ph. 814-865-6435)

Present Use: Non-Government Organizations:

[6] Boeing Houston (POC: David Debrastian, 3700 Bay Area Blvd., Houston TX 77058. Ph. 281-226-4916)
[7] ELORET Corporation (POC: Terrill Buffum, 465 S. Mathilda Ave, Suite 103, Sunnyvale CA 94086. Ph. 408-732-3028)
[8] CalSpan University of Buffalo Research Center - CUBRC (POC: Michael Holden, 4455 Genesse St., Buffalo, NY 14225. Ph. 716-631-6853)
[9] Northrop-Grumman (POC: Les Glatt, Ph. 310-813-1866)
[10] Boeing Huntington Beach (POC: David Yeh, 5301 Bolsa Ave, Huntington Beach CA 92647. Ph. 714-896-1261)
[11] Digital Fusion Solutions (POC: Stacey Rock, Ph. 256-327-0018)
[12] Princeton University (POC: Maria Martin, 247 Western Way, Princeton NJ 08540. Ph. 609-258-7318)

Potential customers include other DoD/DoE installations (including Sandia National Labs), aerospace contractors, and universities (including Georgia Tech) engaged in sponsored research. DPLR has the potential to be the primary aerothermal analysis tool used by both NASA and industry for the Shuttle Orbiter, CEV, and all planetary probes and landers, with users at Boeing, Northrop Grumman, and Lockheed Martin. In addition, DPLR is currently under consideration as the primary aerothermal tool for the Air Force Stability and Transition research program, and if selected will be used by multiple DoD, industrial, and academic installations in support of that objective. Other potential customers include private space launch companies including Space-X and SpaceHab. NASA is actively engaged with ELORET to license and distribute the package. Potential use is Above Average, with about 50 identified current and potential users. Assuming a potential individual user base of ~250 persons at a cost of an industry average $20000 per seat, the monetary assessment of use is approximately $5 million.

Usability of the Software

DPLR is the product of seven years of research and development. During that time parallel programming paradigms have changed repeatedly, and in fact none of the architectures on which the code was originally written remain in production. However, since the underlying algorithm was designed from the start with parallel efficiency and portability in mind, the code remains state-of-the-art. Several utilities are provided to aid in problem setup. Many new users have commented that the learning curve for DPLR is much easier than other related commercial or NASA CFD codes, and new users are typically running production jobs within a few days after gaining access to the software with little or no formal training.

A written user's manual is provided that include examples of problem execution and discuss available options in detail. A set of sample cases is provided that include written instructions and exercise most of the major options of the code. An introductory training DVD is available as part of the distribution. Advanced training courses are under development, and formal training has been given to new users at ARC and JSC. Offsite courses for industrial users are planned at Boeing and Northrop. User support is currently available via email or phone (provided as needed by one of about six persons at NASA Ames and CUBRC), and we are actively engaged with ELORET and the University of Minnesota to host a dedicated user forum page on the WWW.

The end user of DPLR has considerable flexibility in terms of the amount and format of output data, including tracking multiple types of residual information, and a "silent" operation mode in which all standard output is suppressed. In addition, the output of POSTFLOW can be extensively tailored to meet user needs. The user specifies exactly which variables to output in what format over what portion of the geometry. Quantities can be integrated over surfaces, and global minima or maxima can be easily located. Many output formats are supported including the writing of native-format Tecplot® binary datafiles.

An engaged user community has been instrumental to the success of the software, and many of the key features in the code today were suggested by users as part of the feedback process or were in fact developed and implemented by users (now listed as co-developers) and returned to NASA for incorporation into the master source tree.

Quality Factors Considered in Software

Rather than obtaining parallelism by "shoehorning" a heritage algorithm onto a parallel machine, DPLR incorporates a new implicit algorithm (data-parallel line relaxation) designed from the bottom up to be highly efficient and scalable. DPLR supports distributed memory parallelism through the Message-Passing Interface (MPI) industry standard, ensuring portability.

To date DPLR has been successfully run on the CM-5, Cray XMP and T3-D/E series, IBM SP and SMP series, the SGI Origin and Altix, and many types of commodity workstation clusters. DPLR is used heavily on NASA's Columbia supercomputer, and achieves near ideal linear speedup even when large numbers of processors are employed.

DPLR was developed in accordance with standard procedures in place at Ames and within the Space Technology Division. All new versions undergo beta testing prior to their official release. Backward compatibility is always maintained in new releases; users can count on the new executable being seamlessly able to read all file types from previous versions of the code.

The DPLR software is written in Fortran 90 to ensure high efficiency for the numerically intensive computations. Standard "good practice" programming standards are employed to ensure that, e.g., all variables and arrays are declared and that allocated memory is released upon exit. The source code is extensively commented to ensure that DPLR developers present and future can readily modify the source as required. The code also employs a high level of modularity to ensure generality and to minimize multiple instantiations of the same functionality. All datafiles are stored in a platform-independent binary format (XDR), which means that datafiles generated on one machine can be used on another without conversion.

The combination of high parallel efficiency and optimized Fortran core routines allows DPLR to achieve nearly two order of magnitude speedup over the previous state of the art, easily exceeding the original objective of one order of magnitude speedup.

All DPLR source is maintained in a configuration managed database (using standard CVS software). No routines are checked into the database prior to complete validation and testing. A set of benchmark cases has been developed and all new versions are checked against the entire benchmark database prior to release. Release notes for each version include discussion of any physical model updates or bug fixes that may impact execution on the provided benchmark problems, as well as a description of enhancements in the current version. New features in the latest release are highlighted in the provided users manual for easy reference.

Although much of the source code for DPLR is new, it has been written in such a way as to promote both ready maintainability and reuse in future applications. In addition, several of the core routines, have been adapted from legacy applications. Finally, several features, including automatic grid alignment, were developed to be used both as standalone tools and fully integrated into the DPLR software suite.

DPLR incorporates extensive runtime error checking. All common user-induced runtime or setup errors that have been identified are trapped, and an appropriate message is generated to aid in diagnosis. Each release includes additional error checking based on user problem reports. Great care was taken during interface design to ensure that the runtime messages are actually informative and clear, especially as regards possible user errors in setup or execution.

The DPLR package provides the user with a range of runtime options to optimize either performance (at the expense of memory) or memory use (at the expense of performance), ensuring optimum utility on a range of machines from laptops to multi-million dollar dedicated supercomputers.

Efforts to Transfer/Commercialize Software	
Description of Plan/Strategy to Transfer/Commercialize Software	A commercialization agreement is currently being negotiated with ELORET for both distribution and support. To this point the preferred approach has been to negotiate directly with potential customers. Many major aerospace contractors either have the software or are in negotiations to receive it.

NASA Intellectual Property Status/Potential	The DPLR copyright has been assigned to NASA. No patent application has been filed at this time.
Commercialization Potential for the software.	Commercialization potential is somewhat limited by the use restriction to US persons, but strong potential exists within the aerospace and combustion industries. In addition, the core implicit algorithms should be widely commercializable.
Dates Software released for commercial or program use	DPLR was released by the CTO office in June 2005.
List all existing licenses and/or partnership agreements for the software	DPLR has been released to 12 sites. ELORET is in negotiations for a software support and distribution license, and the U. of Minnesota has agreed to host a web-based users group and FAQ.

Innovation (Creative New Features, Solutions, and Achievements)

Rather than obtaining parallelism by "shoehorning" a heritage algorithm onto a parallel machine, DPLR incorporates a novel implicit algorithm designed from the bottom up to be highly efficient and scalable on a variety of parallel architectures. A second novel parallel algorithm, full-matrix data-parallel lower-upper relaxation (FMDP), included as an option, can be very useful for separated flows or unstructured grid applications. Testing has shown that the new methods have reduced the turnaround time for complex reentry flows by nearly two orders of magnitude over the previous state of the art. The implicit algorithms have already been adapted to an unstructured flow solver by the U. of Minnesota, and testing indicates that the method will have high parallel efficiency compared to traditional (e.g. GMRES) methods.

DPLR also provides an industry-leading set of physical models for the solution of hypersonic reacting flows, including thermochemical nonequilibrium, generalized surface chemistry, and support for the pointwise application of any and all initial and boundary conditions, including material properties, turbulence transition maps, and input profiles. There is no COTS equivalent with the combination of parallel performance and physical accuracy provided by DPLR. The modularity of the code facilitates the addition of new physical models as they become available.

The new transport property databases used within DPLR were generated by the developers in support of the code, and are now becoming industry standards for weakly ionized air and CO_2/N_2 plasmas (three journal articles in print and a fourth submitted describe the database).

In addition to its basic execution mode, DPLR has several innovative features that enhance solution quality. One of the most useful is the ability to automatically align the computational grid to the computed shock structure. This results in considerable time savings in terms of grid generation and manipulation and greatly increases the overall solution quality. Since this process has been automated, DPLR is particularly amenable to coupling with optimization routines.

Another important original feature is the fact that parallel decomposition occurs in a manner that is nearly transparent to the end user. The code also permits subfacing, which enables more complex input topologies and grid decomposition strategies. Since the user does not need to concern themselves with the number of blocks that the solution was actually run on, issues like problem setup, post-processing, and changing the parallel decomposition are greatly simplified.

Finally, the extremely powerful post-processor (POSTFLOW) has been written to ensure that the output is fully consistent with the solution, even if the input decks and/or original physical property databases used to generate the solution have been changed or lost.

2005 NASA SOFTWARE OF THE YEAR SUMMARY EVALUATION DOCUMENT FORM

Instructions

The purpose of the Summary Evaluation Document is to provide the Software of the Year (SOY) Panel Members with most of the information necessary to evaluate each nominated software package.

Each Center must submit a Summary Evaluation Document for each software package they nominate. The information provided in the attachment must:

- Fit on six printed pages. A page is a standard 8.5 x 11-inch piece of paper printed in 12 pitch Times Roman font with one-inch margins (top, bottom, and sides). **Note: The SOY Panel Members will only be given the first 6 pages of the Summary Evaluation Document submitted for each software package nominated.**
- Contain all sections of the Summary Evaluation Document form (the evaluation sheet used maps directly to the sections in the Summary Evaluation Document form).
- Be sufficiently focused and accurate to allow the SOY Panel Members to easily understand and score the nominated software. Please use the Glossary for an explanation of terms used in these guidelines and in the evaluation sheet.

There are eight sections on the evaluation sheet and eight corresponding sections in the Summary Evaluation Document form as follows:

Section	Title	Required Information
1. Refer to the glossary in Appendix I for a definition of terms used. 2. For Sections III, IV, V, VI, and VII, use as much space as needed to describe the areas in the Summary Evaluation Document form, however, do not exceed the 6-page limit on the total Summary Evaluation Document form.		
I	Identification Information	Provide: • Software title, same as that used in Form 1329 (Space Act Award Application). • NASA case number assigned during the processing of the NASA Disclosure of Invention and New Technology (Including Software) Form 1679, and • Responsible Center(s) which includes the Center sponsoring the software nomination for SOY award and all other Centers involved in developing the software.
II	Software's Developmental Status	Provide the current Technology Readiness Level (as defined in Appendix II) of the software. If the level is 6 or less the software will be automatically excluded from SOY competition.
III Part A	NASA Mission Significance and Impact	Describe the significance and impact (see definitions of significance and impact in the SOY Glossary) the software has on NASA's mission. Identify: • NASA Headquarters programs, projects and technologies that are being directly supported by this software.

Section	Title	Required Information
1. Refer to the glossary in Appendix I for a definition of terms used. 2. For Sections III, IV, V, VI, and VII, use as much space as needed to describe the areas in the Summary Evaluation Document form, however, do not exceed the 6-page limit on the total Summary Evaluation Document form.		
		• Client group. • Why the software is significant in the technology areas? • The software's impact in these areas.
III Part B	Science, Technology, & Industry Significance and Impact	Describe the significance and impact the software has on science, technology, & industry beyond direct support to NASA's missions (e.g., biotechnology, medicine, education, etc.). This refers to the adaptation of NASA mission technologies to secondary technology application areas for clientele different than those originally intended. These technology areas are known as horizontal technologies (see glossary). Identify: • The sciences and/or technologies that are being directly supported by this software. • Client group. • Why the software is significant in the horizontal technology application areas. • The software's impact in these areas.
III Part C	Impact on the Quality of Human Life	Describe the significance and impact the software has on the quality of human life. Consider such things as: • Intellectual impact • Environmental impact • Energy conservation impact • Tool to help improve human understanding of life • Health and safety impact • Improvement in processes such as: administrative, technical, research, educational, etc.
IV	Extent of Current and Potential Use	Describe the extent to which the software is supporting or has potential to support government & private sector efforts. For present use identify: • Federal, state, and/or local governments using the software. • Non-government (private sector) organizations using the software. • Points of contact for each government and non-government organization using the software, including name, address, and phone number. For potential use identify:

Instructions

Section	Title	Required Information
1. Refer to the glossary in Appendix I for a definition of terms used. 2. For Sections III, IV, V, VI, and VII, use as much space as needed to describe the areas in the Summary Evaluation Document form, however, do not exceed the 6-page limit on the total Summary Evaluation Document form.		
		• Federal, state, and/or local governments that may make use of the software. • Non-government (private sector) organizations that may make use of the software. • Where and how the software's sponsoring organization intends to try to expand the use of the software. For both current and potential use identify the level of use (modest, average, above average and excellent as defined in the glossary of these instructions).
Creativity		Components used to evaluate software creativity on the software evaluation sheet are: – The usability of the software (approximately 10 % of the creativity score) – The quality of the software package (approximately 40% of the creativity score) – The efforts made to commercialize the software (approximately 10% of the creativity score), and – Innovation produced in the development of the software (approximately 30% of the creativity score).
V	Usability of the Software	Describe key factors, which make the software easy for the end user to use. Specifically address: • Ease of use features that help the end-user understand system displays, input requirements, and outputs. • Technical support provided for problem consultation, trouble-shooting, debugging, fixes, maintenance, and enhancements. • Documentation available including help functions. • Training available. Describe the courses to include media used (e.g., classroom, web, videos, etc.) target audience and schedule for the next 12 months.
VI	Quality Factors Considered in Developing the Software	Provide the justification used for selecting each of the following: • Architecture (e.g., Object oriented, functional decomposition, etc.)

<table>
<tr><th>Section</th><th>Title</th><th>Required Information</th></tr>
<tr><td colspan="3">1. Refer to the glossary in Appendix I for a definition of terms used.
2. For Sections III, IV, V, VI, and VII, use as much space as needed to describe the areas in the Summary Evaluation Document form, however, do not exceed the 6-page limit on the total Summary Evaluation Document form.</td></tr>
<tr><td></td><td></td><td>• Programming language(s) used
• Operating environment (e.g., operating system(s), hardware platform(s), web interactive interface(s), etc.)
Furthermore, describe the quality factors that were addressed in developing the software and the tradeoffs made between each factor listed:
• Reliability
• Function
• Performance - to include a description of the performance objectives and technical performance measures that were used. Also indicate if the original performance objectives were achieved.
• Reuse
• Maintainability
See the glossary included in these instructions for definitions of each of the above terms.</td></tr>
<tr><td>VII</td><td>Efforts to Transfer / Commercialize Software</td><td>Identify efforts made to transfer or commercialize the software including:
• Plan/strategy to transfer or commercialize the software. This should include, but is not limited to, establishing licensable IP, marketing the software for commercial use and licensing, and creating NASA/industry partnerships.
• IP status and potential of the software, including efforts to establish rights in inventions, copyrights and trademarks that are licensable by NASA.
• Commercialization potential assessed, including the identification of key market factors, commercial needs, and the suitability of the software.
• Date(s) the software was released for commercial use in accordance with NPD/NPG 2210.
• List all existing IP licenses associated with the software in a commercial environment or NASA/industry partnership agreements for the development /commercialization of the software.</td></tr>
<tr><td>VIII</td><td>Innovation</td><td>Describe the extent of innovation (newness, originality, and/or uniqueness) involved in developing the software. Specifically address:</td></tr>
</table>

Instructions

<table>
<tr><th>Section</th><th>Title</th><th>Required Information</th></tr>
<tr><td colspan="3">1. Refer to the glossary in Appendix I for a definition of terms used.
2. For Sections III, IV, V, VI, and VII, use as much space as needed to describe the areas in the Summary Evaluation Document form, however, do not exceed the 6-page limit on the total Summary Evaluation Document form.</td></tr>
<tr><td></td><td></td><td>• The extent to which the software is a redevelopment of COTS equivalent software available in the market. If COTS equivalent software exists, state why the COTS was not used and why the equivalent software was developed.
• Improvement/non-trivial modification to the state of the art that was made in developing the software.
• Any advances in the state-of-the-art achieved by the software.
• Any ground-breaking/original software technologies such as new or novel methods, techniques, languages, processes, etc.</td></tr>
</table>

2006 NASA SOFTWARE OF THE YEAR SUMMARY EVALUATION DOCUMENT FORM
APPENDIX I
GLOSSARY

Advances the State-of-the-Art: Software that significantly improves or updates currently existing concepts, operating environments, development tools, languages or new processes.

Assessment of Use: An evaluation of the extent of present use of the software and of potential use/marketability of the software. Levels of use or potential use may be defined as follows:

- Modest: less than $1.0 million of useful value.
- Average: between $1.0 million and $10 million of useful value.
- Above Average: between $10 million and $100 million of useful value.
- Excellent: over $100 million of useful value.

Copyright: A government issued grant of exclusive right to an author for an original work that is fixed in a tangible medium of expression, such as software. This right includes the right to exclude others from copying, distributing, and from developing other software derived from the copyright protected software.

COTS (Commercial Off The Shelf) Equivalent SW Available on Market: Are there any software products on the market that are equivalent in functionality and capability to the nominated software product

Creativity: See innovation. Components used to evaluate software creativity on the software evaluation sheet are:

– The usability of the software (approximately 10 % of the creativity score)
– The quality of the software package (approximately 40% of the creativity score)
– The efforts made to commercialize the software (approximately 10% of the creativity score), and
– Innovation produced in the development of the software (approximately 30% of the creativity score).

Development Status: The current Technology Readiness Level (TRL) of the software package. If the software is rated between 1 and 6, it is automatically disqualified from further SOY competition. The definitions of the TRL levels are found in Appendix II.

Documentation Quality: The degree to which published operating procedures, system functional descriptions, and technical specifications are understandable and useful.

Ease of Use: The end user's perspective of how effortless the system is to interact with and understand. This includes several user related issues such as:

– User system interface (e.g., a graphical user interface (GUI)) and the mechanisms (menus, icons and buttons) by which the user exercises the system functions,
– User support provided, and
– Flexibility in changing the content and format of system outputs (reports, displays, and other output).

2006 NASA SOFTWARE OF THE YEAR SUMMARY EVALUATION DOCUMENT FORM
APPENDIX I
GLOSSARY

Efforts to Commercialize Software: Patent council determination that the software may be licensable, patents, copyrighted material, trade secrets, inventions, trademarks and other knowledge that is the basis for commercializing the software.

Function: How closely the system processes match the end user's requirements. Also, refers to verification of the software program with regard to its correctness in meeting the requirements/specifications.

Ground Breaking/Original: Software applications whose functionality never existed before. This item refers to the development of new software technologies such as new languages, methods, techniques and processes.

Government Potential Use: The likelihood that the currently operational NASA software may be utilized in support of other government agencies (federal, state, or local).

Government Present Use: The extent of current federal, state, and/or local government utilization of the currently operational NASA software.

Horizontal Technology: A Technology in one technology area of application that is adapted to a different area of application.

Impact: The effect of the software on the program, and/or project. Examples of impact include: cost and timesavings, increased productivity, reduced risk, and increased security and safety,

Improvement/Non-Trivial Modification: New software or any pre-existing software modified by more than a trivial variation or improvement. A trivial variation or improvement includes minor code improvements that do not materially alter the software's operation.

Innovation: Producing meaningful new ideas, forms, methods, techniques, processes, systems, and interpretations or analogies. Also, using new knowledge, ideas, and/or inventions to create new products or services. Components used to evaluate software creativity on the software evaluation sheet are:

– Whether or not there is equivalent COTS software available,
– Improvement/non-trivial modification of previously existing software,
– Advances in the state-of-the-art, and
– Groundbreaking/original effort.

Invention: Any new idea, concept, technique, device, or process that has not yet been commercialized.

Justification for selecting technology and/or approach chosen: This justification is concerned with use of effective architecture(s), languages and tools. What efforts were made to select an architecture that would assure the optimal technological approach? For example:

– What was the architecture (Object-oriented, Function-based, etc) chosen and why?

– What language(s) (such as 4GLs or specialized languages) was chosen and why?

Maintainability: The ease and cost-effectiveness of system trouble-shooting, fixes, upgrades, and enhancements to meet changing system requirements.

NASA Case No: The number used in Form 1329 and is assigned by the Center Patent Attorney during processing of the New Technology Disclosure Form 1679.

Non-Government Potential Use: The likelihood that the currently operational NASA software may be utilized in the support of industry and non-profit sectors.

Non-Government Present Use: The extent of current utilization by industry and/or non-profit sectors of the currently operational NASA software.

Other Science and Technologies: Horizontal or crosscutting technology areas (e.g., Biotechnology, Communications, Construction, Education, Environment, Information Technology, Manufacturing, Materials, Medicine, etc) and secondary uses of the technology:
– Where the user(s) is not necessarily part of the clientele group for whom the application was originally developed.
– Whose application extends outside of NASA's mission support.

Patent: A government grant issued to an inventor or applicant for an invention that gives the inventor or applicant the right to exclude others from making, using, selling, or importing the patented invention.

Performance: The efficiency and effectiveness of the software system operation, in terms of responsiveness, throughput, cost and other technical performance measures. Response is a measure of how quickly and effectively the system reacts to a user's interaction with the system. Throughput is a measure of the computational work (based on workload characterization) accomplished by the system (software and hardware) within a specified time. The technical performance measures vary from system to system.

Portability: The extent of compatibility of the software with different operating system environments.

Quality: The extent of the superiority or excellence of the software measured by factors such as: how correctly the software performs the functions for which it was designed; system performance; system reliability; maintainability; and reuse of design, specifications and code.

Reliability: A measure of the probability that a system is operating satisfactorily at a given time. Also, refers to failsafe features built into the application.

Responsible Center: this is the sponsoring Center of the software nominated for the Software of the Year (SOY) Award.

2006 NASA SOFTWARE OF THE YEAR SUMMARY EVALUATION DOCUMENT FORM
APPENDIX I
GLOSSARY

Reuse: The extent to which the design, specifications, and/or source lines of certified software code of the system being considered for the SOY Award has been structured to facilitate adoption into systems to be developed in the future. Also, the extent to which previous designs, specifications, and/or source lines of certified software code have been incorporated into the system being considered for SOY award.

Science and Technology Significance: The extent of impact the software has on NASA's missions and/or the impact of the software on other science and Technology. See "Other Science and Technology" for further definition in this area.

Significance: Why something stands out or is important. Examples include: unique or greatly improved processes or products; functions, analytical tools and models that enable the development of systems or enable the execution of missions; and new and unique product that has a high probability of commercial success.

Software Class (from NPR 7150.2, NASA Software Engineering Requirements):

Class A Human Rated Software Systems	Applies to all space flight software subsystems (ground and flight) developed and/or operated by or for NASA to support human activity in space and that interact with NASA human space flight systems. Space flight system design and associated risks to humans are evaluated over the program's life cycle, including design, development, fabrication, processing, maintenance, launch, recovery, and final disposal. Examples of Class A software for human rated space flight include but are not limited to: guidance; navigation and control; life support systems; crew escape; automated rendezvous and docking; failure detection, isolation and recovery; and mission operations.
Class B Non-Human Space Rated Software Systems	Flight and ground software that must perform reliably in order to accomplish primary mission objectives. Examples of Class B software for non-human (robotic) spaceflight include, but are not limited to, propulsion systems; power systems; guidance navigation and control; fault protection; thermal systems; command and control ground systems; planetary surface operations; hazard prevention; primary

	instruments; or other subsystems that could cause the loss of science return from multiple instruments.
Class C Mission Support Software	Flight or ground software that is necessary for the science return from a single (non-critical) instrument or is used to analyze or process mission data or other software for which a defect could adversely impact attainment of some secondary mission objectives or cause operational problems for which potential work-arounds exist. Examples of Class C software include, but are not limited to, software that supports prelaunch integration and test, mission data processing and analysis, analysis software used in trend analysis and calibration of flight engineering parameters, primary/major science data collection and distribution systems, major Center facilities, data acquisition and control systems, aeronautic applications, or software employed by network operations and control (which is redundant with systems used at tracking complexes). Class C software must be developed carefully, but validation and verification effort is generally less intensive than for Class B.
Class D Analysis and Distribution Software	Non-space flight software. Software developed to perform science data collection, storage, and distribution; or perform engineering and hardware data analysis. A defect in Class D software may cause rework but has no direct impact on mission objectives or system safety. Examples of Class D software include, but are not limited to, software tools; analysis tools, and science data collection and distribution systems.
Class E Development Support Software	Non-space flight software. Software developed to explore a design concept; or support software or hardware development functions such as requirements management, design, test and integration, configuration management, documentation, or perform science analysis. A defect in Class E software may cause rework but

	has no direct impact on mission objectives or system safety. Examples of Class E software include, but are not limited to, earth science modeling, information only websites (non-business/information technology); science data analysis; and low technical readiness level research software.
Class F General Purpose Computing Software (Multi-Center or Multi-Program/Project)	General purpose computing software used in support of the Agency, multiple Centers, or multiple programs/projects, as described for the General Purpose Infrastructure To-Be Component of the NASA Architecture, Volume 5 (To-Be Architecture), and for the following portfolios: voice, wide area network, local area network, video, data centers, application services, messaging and collaboration, and public web. A defect in Class F software is likely to affect the productivity of multiple users across several geographic locations, and may possibly affect mission objectives or system safety. Mission objectives can be cost, schedule, or technical objectives for any work that the Agency performs. Examples of Class F software include, but are not limited to, software in support of the NASA-wide area network; the NASA Web portal; and applications supporting the Agency's Integrated Financial Management Program, such as the time and attendance system, Travel Manager, Business Warehouse, and E-Payroll.
Class G General Purpose Computing Software (Single Center or Project)	General purpose computing software used in support of a single Center or project, as described for locally deployed portions of the General Purpose Infrastructure To-Be Component of the NASA Architecture, Volume 5 (To-Be Architecture) and for the following portfolios: voice, local area network, video, data centers, application services, messaging and collaboration, and public web. A defect in Class G software is likely to affect the productivity of multiple users in a single geographic location or workgroup, but is unlikely to affect mission

	objectives or system safety. Examples of Class G software include, but are not limited to, software for Center custom applications such as Headquarters' Corrective Action Tracking System and Headquarters' ODIN New User Request System.
Class H: General Purpose Desktop Software	General purpose desktop software as described for the General Purpose Infrastructure To-Be Component (Desktop Hardware & Software Portfolio) of the NASA Architecture, Volume 5 (NASA To-Be Architecture). This class includes software for Wintel, Mac, and Unix desktops as well as laptops. A defect in Class H software may affect the productivity of a single user or small group of users but generally will not affect mission objectives or system safety. However, a defect in desktop IT-security related software, e.g., anti-virus software, may lead to loss of functionality and productivity across multiple users and systems. Examples of Class H software include, but are not limited to, desktop applications such as Microsoft Word, Excel, and Power Point, and Adobe Acrobat.

Technical Support: The support available for user assistance, trouble-shooting, fixes, upgrades, enhancements, and documentation.

Technology Commercialization: The process of new technology development through partnerships with government and industry with the objective of creating new products, processes, or services with commercial potential.

Technology Transfer: The process by which technology developed in one organization, in one area, or for one purpose is applied in another organization, in another area, or for another purpose

Technology Readiness Levels (TLR): The level of software system development. There are nine software technology readiness levels, ranging from 1 to 9, associated with the NASA software development life cycle and software having a TRL of 6 or less is automatically disqualified from the Software of the Year competition.

Software Title: the software title should be the same as that used in Form 1329 (Space Act Award Application).

2006 NASA SOFTWARE OF THE YEAR SUMMARY EVALUATION DOCUMENT FORM
APPENDIX I
GLOSSARY

Understandability: The degree to which the end-user can easily grasp the conceptual operation of the software (i.e., the system architecture). For example, can the end-user easily understand the system displays and outputs?

Usability: How well the user can apply the system functions to his/her needs. The software system usability attributes include understandability, ease-of-use, availability of technical support, quality end-user documentation, and availability of training.

2006 NASA SOFTWARE OF THE YEAR SUMMARY EVALUATION FORM
APPENDIX II
TECHNOLOGY READINESS LEVELS APPLIED TO SOFTWARE

TRL 9: Actual system "flight proven" through successful mission operations

Thoroughly debugged software. Fully integrated with operational hardware/software systems. All documentation has been completed and users have successful operational experience. Sustaining software-engineering support in place. Actual system fully demonstrated.

TRL 8: Actual system completed and "flight qualified" through test and demonstration (Ground or Flight)

Thoroughly debugged software. Fully integrated with operational hardware and software systems. Most user documentation, training documentation, and maintenance documentation completed. All functionality tested in simulated and operational scenarios. V&V completed.

TRL 7: System prototype demonstration in a relevant environment

Most of the software is functionality available for demonstration and test. Well integrated with operational hardware/software systems. Most software bugs removed. Limited documentation available.

TRL 6: System/subsystem model or prototype demonstration in a relevant environment (Ground or Space)

Prototype implementations if the software is on full-scale realistic problems. Partially integrated with existing hardware/software systems. Limited documentation available. Engineering feasibility fully demonstrated.

TRL 5: Component and/or breadboard validation in relevant environment

Prototype implementations. Experiments with realistic problems. Simulated interfaces to existing systems.

TRL 4: Component and/or breadboard validation in laboratory environment

Standalone prototype implementations. Experiments with full-scale problems or data sets.

TRL 3: Analytical and experimental critical function and/or characteristic proof-of-concept

Limited functionality implementations. Experiments with small representative data sets. Scientific feasibility fully demonstrated.

TRL 2: Technology concept and/or application formulated

Basic principles coded. Experiments with synthetic data. Mostly applied research.

TRL 1: Basic principles observed and reported

Basic properties of algorithms, representations & concepts. Mathematical formulations. Mix of basic and applied research.